

# Computer vision Seminar

수원대학교  
Data Network Analysis 

---

데이터과학부  
정인호

2023.03.09

# Week7) Semantic Segmentation

---

1. 마지막 코드 실습
2. Semantic Segmentation
3. FCN(Fully Convolution Network)
4. U-Net & 마지막 과제 설명

# Week7) Semantic Segmentation

---

## 1. 마지막 코드 실습

```
def ComputerVision_Seminar(방학, 과제, 시간, 노력):  
  
    ...  
  
    return 깃허브, 코딩실력, 이론지식
```

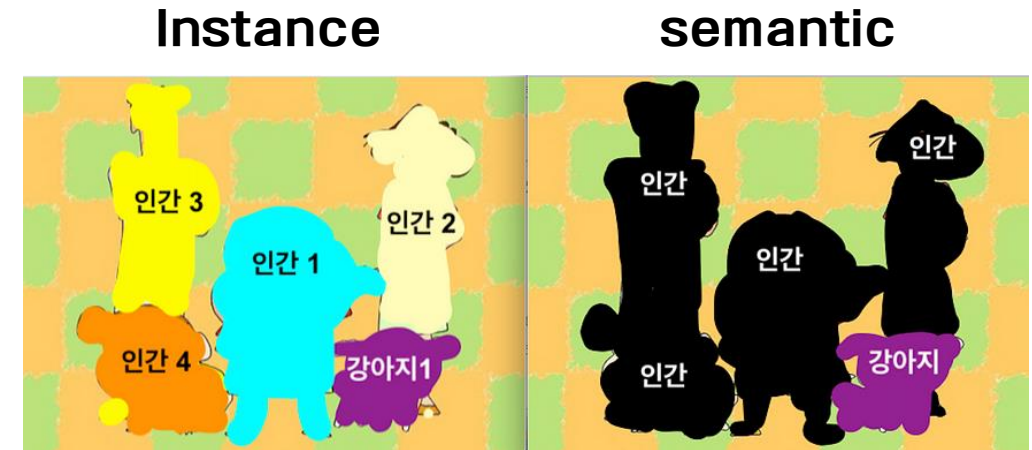
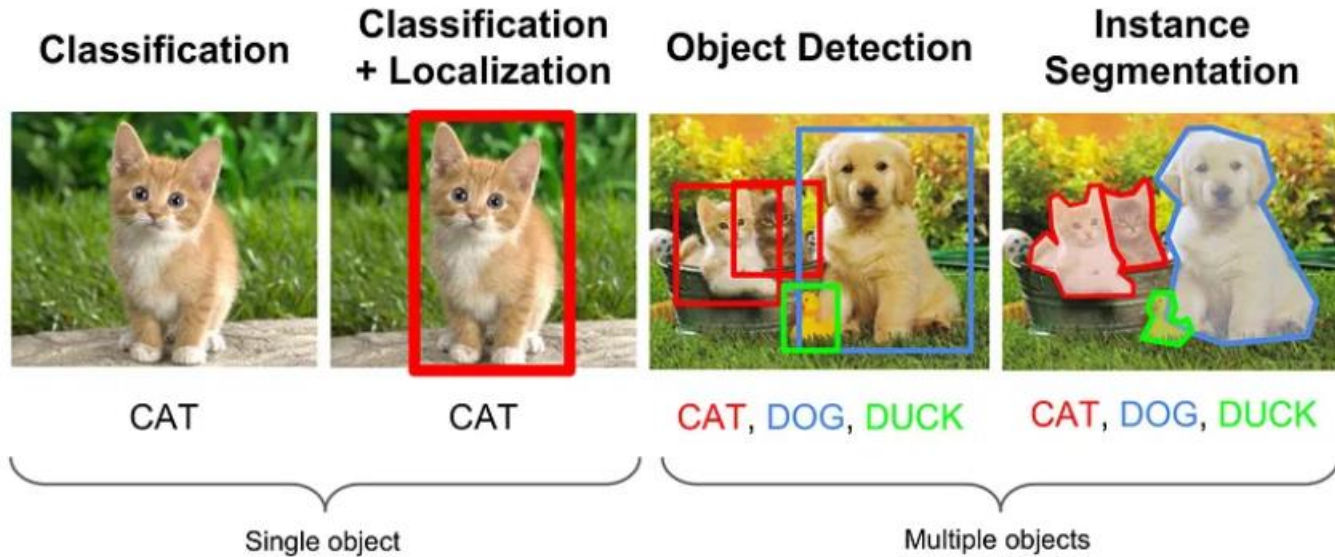
# Week7) Semantic Segmentation

---

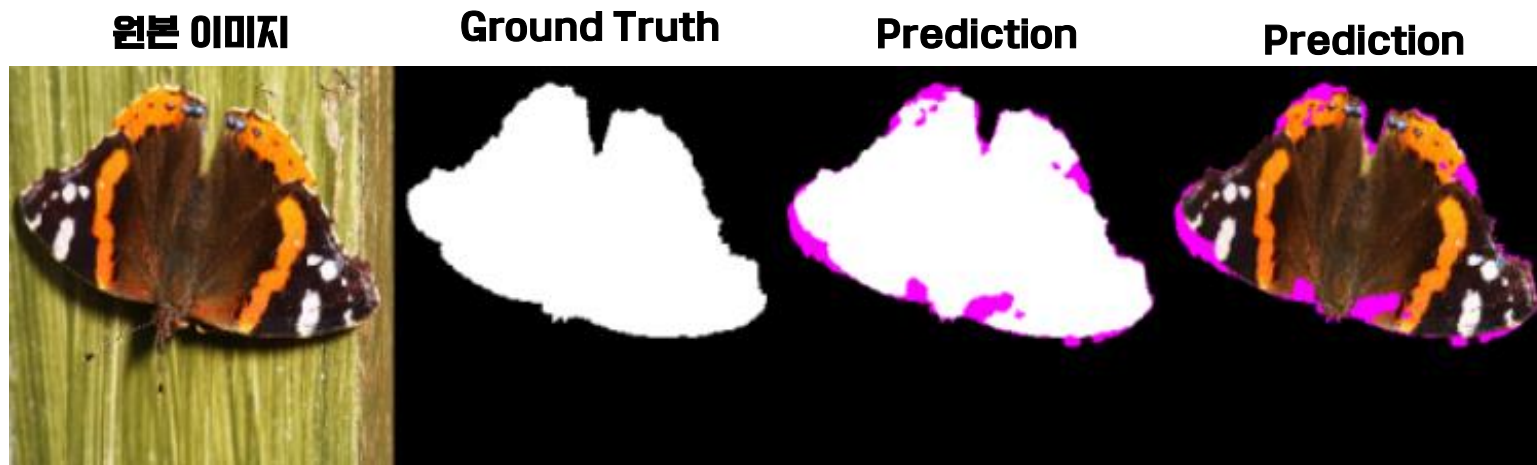
## 2. Semantic Segmentation



## 2. Sematic Segmentation



## 2. Sematic Segmentation



# Week7) Semantic Segmentation

---

## 3. FCN



# 3. FCN(Fully Convolution Network)(2014)

## Abstract

FCN은 end-to-end, pixels-to-pixels 학습이 되는 convolutional network입니다. 핵심 아이디어는 임의의 크기로 입력 값을 받고, 그에 해당하는 출력값을 생성하는 'fully convolutional network' 입니다.

AlexNet, GoogLeNet, VGGnet과 같은 classification 신경망을 사용하고, 이들을 segmentation task에 맞게 fine-tuning 합니다. 그리고 나서 shallow의 정보와 deep의 정보를 결합하는 새로운 구조를 정의합니다.

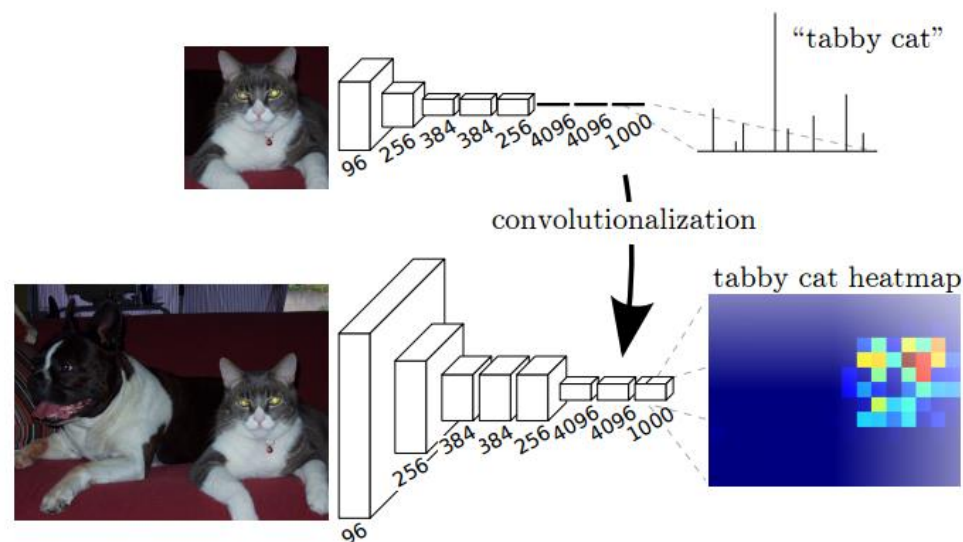


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.



# 3. FCN(Fully Convolution Network)(2014)

segmentation을 위한 새로운 fully convolutional network를 정의합니다. layer를 결합하고 출력값의 공간적인 정보를 개선합니다.

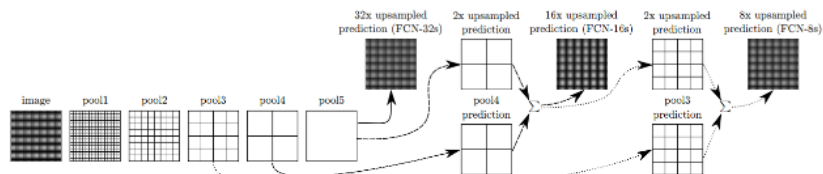


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.

중앙일보

fully convolutionalized classifiers는 segmentation을 위해 finetuning할 수 있습니다. 높은 standard metric을 얻더라도, 이 출력값은 coarse를 만족하지 않습니다. 아래 그림에서 FCN-32s를 보면, 픽셀들이 뭉쳐져 있는 것을 확인할 수 있습니다. 마지막 prediction 레이어에서 32 pixel stride은 upsampled output에서의 scale을 제한합니다.

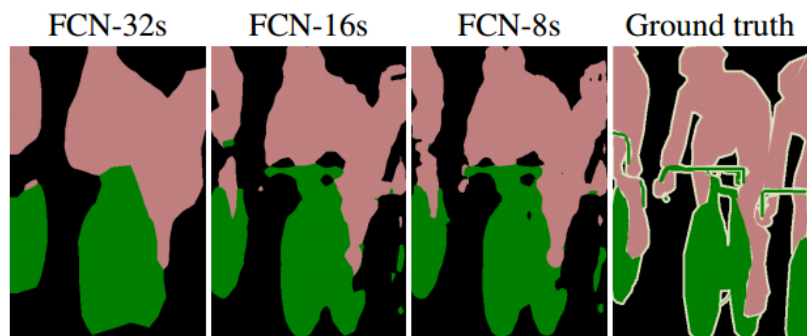


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

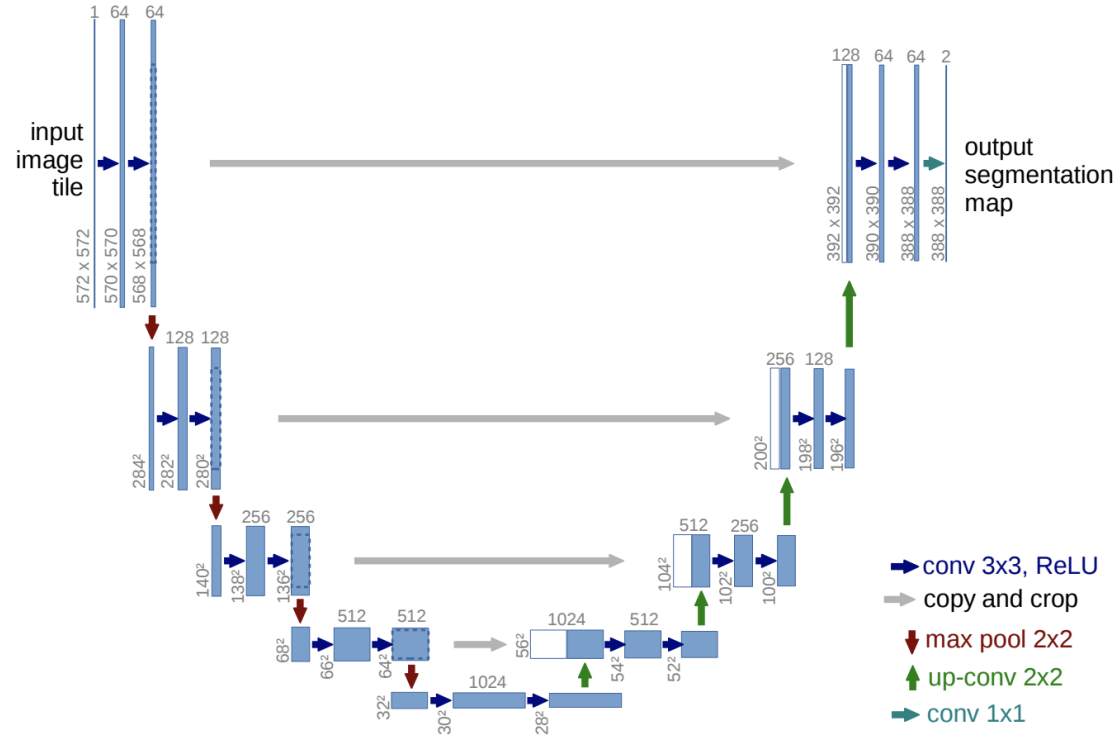
# Week7) Semantic Segmentation

---

## 4. U-Net



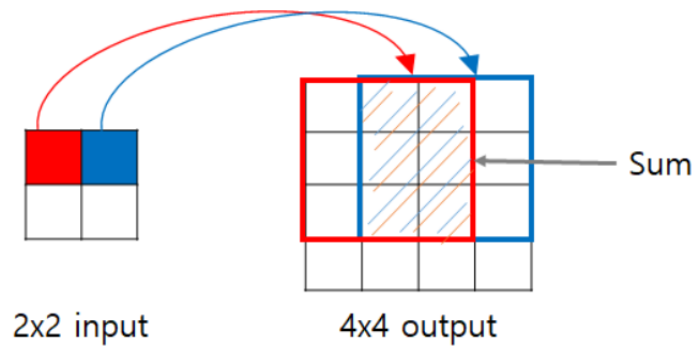
# 4. U-Net



**Fig. 1.** U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

# 4. U-Net

## Transposed Conv 2d



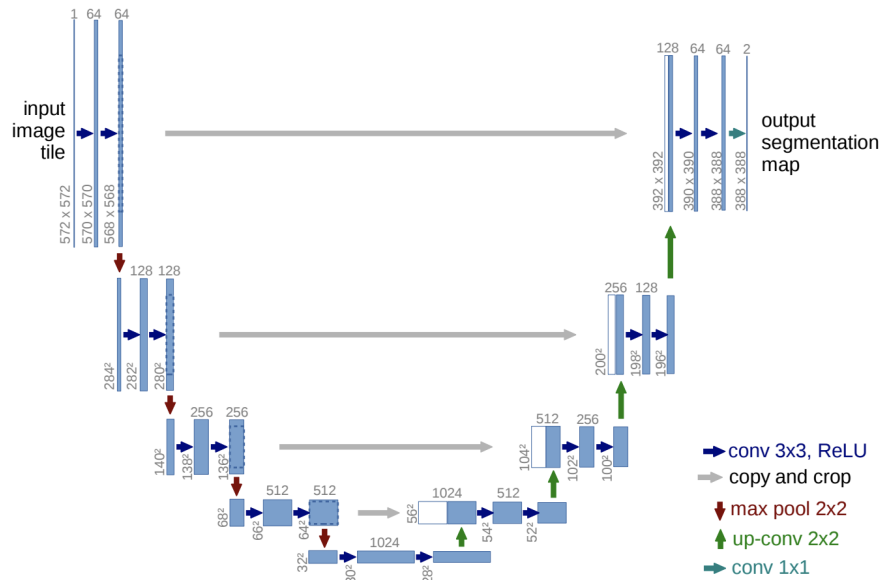
input과 output의 위치를 바꿔서 생각해보자. 위 그림에서 input의 빨간색 원소는 output의 빨간 박스 안에 있는 원소들과 연관이 있다. 마찬가지로 input의 파란색 원소는 output의 파란색 박스 안에 있는 원소들과 대응한다.

Transposed Convolution을 계산하는 방법은 이렇다. input의 빨간색 원소를 3x3 kernel에 곱해서 output의 대응하는 자리에 집어넣는다. 같은 방법으로 input의 파란색 원소를 3x3 kernel에 곱해서 output의 대응하는 위치에 집어넣는다. 이 때 output에 겹치는 구간(빗금 표시)이 발생하는데, 겹치는 부분의 값은 모두 더 해준다. input의 나머지 원소에 대해서도 동일한 방법으로 계산한다.

## 이미지 보관법

# 4. U-Net

## U-net 핵심정리



**Fig. 1.** U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

# 4. 마지막 과제 설명

## Last 과제

최종 코드까지 마무리하여 깃허브를 정리한 뒤, 저한테 링크를 보내주세요.

```
def ComputerVision_Seminar(방학, 과제, 시간, 노력):
```

```
...
```

```
return 깃허브, 코딩실력, 파이토치, 이론지식, 흥미
```

[발표자 정인호 깃허브](#)

# 다들 수고 많으셨습니다. 😊

---

Unofficial ending